DEPLOYMENT DAY 06: PREPARATION AND STAGING ENVIRONMENT SETUP

Objective:

The objective of Day 6 is to prepare the marketplace for deployment by setting up a staging environment, configuring a hosting platform, and ensuring the application operates seamlessly in a production-like environment. This involves setting up environment variables, testing, and documenting all steps to ensure readiness for deployment.

Steps Taken For Deployment Preparation

1.Hosting Platform Selection

We chose Versal as the hosting platform for its simplicity and efficient deployment process. The following configurations were made:

Connected GitHub Repository: The project repository was linked to Versal for continuous integration and deployment.

Build And Deployment Settings: Configured to ensure successful staging builds.

2.Environment Variables Configuration

Sensitive data such as API keys and project configurations were managed securely.

Such as:

NEXT_PUBLIC_SANITY_PROJECT_ID=your_project_id
NEXT_PUBLIC_SANITY_DATASET=production

API_KEY=your_api_key

Environment variables securely added through the Vercel dashboard.

3.Staging Environment Setup

Deployed the application to a staging environment to simulate production. This involved:

- Validating succesful build without error.
- Verifying basic functionality, such as the homepage, navigation, and interations.

4.Staging Environment Testing

1.Functional Testing:

Verified all workflows (e.g., product listing, cart operations, etc.) in the staging environment.

Test Case ID	Description	Steps	Expected Result	Actual Result	Status	Remarks

TC001	Validate product listing	Open product page	Products display correctly	Products display correctly	Passed	No issues found
TC002	Validate product listing	Disconnect API > Refresh page	Show fallback message	Fallback message shown	Passed	Handled gracefully
TC003	Validate cart functionality	Add item to cart > Check cart	Cart updates correctly	Cart updates correctly	Passed	Works as expected

2.Performance Testing:

Analyzed speed and responsiveness using GTmetrix and Lighthouse . Metrixs such as load time, interactivity, and layout shifts were documented. Reports from GTmetrix provided detailed insights into performance bottlenecks and optimization opportunities.

Security Testing

Validated the following:

- Input fields to prevent SQL injection and XSS attacks.
- Secure Handling of API keys.
- HTTPS implementation.

Expected Output

- A fully functional staging environment hosted on Versal.
- Environment variables securely configured.
- Test case and performance reports uploaded to GitHub.
- A professional README.md summarizing all project activities.

Presented By**: RIMSHA SHEIKH**

Date: 22-January-2025